REMARKS

Claims 1-22 were pending in the subject case. In the Office Action dated 2/25/2003 ("Final Action"), Claims 1, 11, and 19 were rejected, and Claims 2-10, 12-18, and 20-22 were objected to. Claims 1-22 remain pending as previously presented. In view of the arguments set forth below, it is respectfully submitted the Claims are in condition for allowance.

ARGUMENTS

1. Claims 1, 11, and 19 were finally rejected under 35 USC §102(b) as being anticipated by U.S. Patent No. 6,338,133 to Schroter ("Schroter"). This rejection is respectfully traversed. The Examiner's arguments set forth in the Final Action are addressed below in turn.

First, a general comparison of Applicants' inventive system and method as compared to that of Schroter is set forth for discussion purposes. This comparison will then be directed to the language of Applicants' various independent Claims.

Schroter describes a data processing system that includes execution circuitry to execute instructions received from a dispatch unit 20. This execution circuitry comprises a branch processing unit (BPU) 18, a fixed-point unit (FXU) 22, a load/store unit (LSU) 28, and a floating-point unit (FPU) 30. (Schroter column 5 lines 11-14.) Each of the FXU, LSU, and FPU execution units has a respective instruction queue to store instructions waiting to be executed by that execution unit. (See

Schroter Figure 4A, queues 407A-407C.) Each of these queues is associated with a respective threshold value. When dispatch unit 20 encounters a speculative branch instruction, that instruction will not be provided to any of the execution units if the number of instructions stored within one or more of these instruction queues exceed the respective threshold value. In one embodiment, the speculative instruction will not be dispatched if any of the threshold values are exceeded. (See Figures 4A.) In other embodiments, the instruction will be dispatched if some sub-set of the threshold levels is met. This functionality is intended to minimize processing delays that might occur because of branch mispredictions, and to thereby keep the various execution units executing as fast as possible.

In contrast to Schroter system and method, Applicants' disclose and claim a pipeline control system and method that allows a user to selectively throttle the number of instructions that begins concurrent execution within a predetermined period of time. This can best be understood in reference to Applicants' Figures. In Applicants' Figure 2, the pipeline is executing at full speed such that exactly six instructions undergo concurrent execution within six clock cycles, which is considered the "predetermined time period" in this example. This time period is selected because Applicants' exemplary pipeline includes six execution stages, each being one clock cycle long. In Figure 10, the maximum pipeline execution rate illustrated in Figure 2 is throttled such that only a single instruction begins execution within the predetermined time of six clock cycles. For example, P1 is the only instruction that begins execution during clock cycles 8-13. Similarly, P2 is the sole instruction beginning execution during clock cycles 9-14, and so on. Applicants' Figures 11

through 17 provide additional examples of this throttling mechanism. For instance, in Figure 11, exactly two instructions are allowed to begin concurrent execution within the predetermined time period of six cycles, and so on.

The fundamental differences between Schroter and Applicants' invention are now considered with respect to Applicants' Claim language, and the arguments set forth by the Examiner in the Final Action.

Applicants' Claim 1 includes the following:

> "a first storage device to receive and to store a programmable count value indicative of a predetermined number of instructions; and
> a logic sequencer coupled to said first storage device to receive said programmable count value, and in response thereto, to generate a pipeline control signal provided to the instruction pipeline to cause the instruction pipeline to receive, and to initiate concurrent execution on, the predetermined number of the instructions in the predetermined period of time."

These aspects of Claim 1 are not taught by Schroter for at least the following reasons:

A.) Schroter does not teach a programmable count value that indicates a number of instructions on which execution is initiated in a predetermined period of time.

The Examiner states that the Schroter threshold values teach this aspect of Applicants' invention. However, as discussed above, each Schroter threshold value controls the number of instructions that may be stored within a respective queue before speculative instruction dispatch is delayed. For example, according to one embodiment, if a threshold value is set to "two" for a given queue, a speculative instruction will not be dispatched by dispatch unit 20 until that queue stores fewer than two instructions. When the number of

instructions within the queue drops to fewer than two, the speculative instruction will be stored in the appropriate one of queues 407A – 407C to await execution.

As can be appreciated from the foregoing discussion, setting the threshold value of a queue to "two" does not necessarily control the number of instructions stored within that queue, since non-speculative instructions continue to be dispatched without regard to threshold values. This threshold value may only delay instruction dispatch when a speculative instruction is encountered. Moreover, this threshold value certainly does not control how many instructions *begin execution* within the pipeline within a predetermined time. This can be appreciated by further considering the Schroter circuitry as follows.

The Schroter processor includes multiple execution units FXU, LSU, and FPU. For discussion purposes, it will be assumed that these execution units comprise the Schroter pipeline, as the Examiner asserts in the Final Action. (Final Action, page 4 lines 14-16.) Schroter states that each of these execution units executes one or more instructions during each processor cycle. (Schroter column 5 lines 15-18.) It appears that each of these execution units continue retrieving instructions from a respective queue as fast as possible, and without any regard to how fast the instructions are *added* to the queue by dispatch unit. Since threshold levels only affect on how fast speculative instructions *are added* to the queues, the threshold levels certainly do not directly control the execution rates. At most, the threshold levels may have some indirect affect on execution rates when a relatively large percentage of speculative instructions are encountered within the instruction stream, perhaps causing the load level of one or more of queues 407A-407C to temporarily decrease. The exact affect this situation may have on execution rates is hard to predict, and is not discussed in Schroter. Certainly, Schroter does not teach a system wherein setting one or more of the threshold level to any particular number "N" will dictate that "N" instructions begin execution in some predetermined period of time, as is claimed by Applicants' Claim 1.

The Examiner maintains that the threshold levels do directly correspond to the number of instructions being executed by the execution units. In support of this assertion, the Examiner cites the case wherein a threshold value is set to zero. As best understood, the Examiner is asserting that if all threshold values are set to the predetermined number of "zero", the execution rate will drop to this predetermined number of "zero". This is not understood. First, consider that non-speculative instructions will continue to be dispatched to queues 407A-407C, so that dispatch and execution of these instructions will continue completely unaffected by the threshold rates. This could continue indefinitely so long as no speculative instructions are encountered in the instruction stream. Thus, setting the threshold values to "zero" does not necessarily mean that "zero" instructions will be executed in some predetermined period of time. Moreover, when a speculative instruction is encountered, it will be provided to the appropriate execution unit when all load levels drop to zero. It is hard to determine what the execution rate will be at that time, since the other execution units may still be processing other instructions. Whether or not the execution rate will be "zero" for some period of time is hard to predict, but it certainly cannot be controlled, since it most certainly will depend on the types of instructions in the instruction stream. This special case scenario is not discussed in any way in Schroter, and the Examiner's assumption that the Schroter system will behave in any particular way in this instance is merely a supposition. Moreover, employing this scenario in Schroter would seem to defeat the very purpose of the invention, which is to ensure that the execution units don't have to "wait for work", but instead, have an adequate supply of instructions stored within the instruction queues. (See Schroter column 8 lines 23-27.)

Next, the Examiner sets forth the additional argument that once the queues are full, the pipeline cannot continue. This, therefore, indirectly controls how many instructions are moving through the larger pipeline. This argument is not understood. If the queues become full, they are capable of supplying instructions to the various execution units without delay.

The execution units will presumably process instructions at the maximum possible rate, which appears to be one or more instructions during each processor cycle. It is unclear how this special case teaches Applicants' invention for setting a programmable value to a predetermined number to ensure that exactly that predetermined number of instructions begins execution with the pipeline in some predetermined period of time.

For at least the foregoing reasons, Schroter does not teach or suggest Applicants' invention of Claim 1.

B. Schroter does not teach a system wherein within some predetermined period of time, the number of instructions that commence can be precisely controlled by a count value.

Applicants' invention of Claim 1 provides an instruction pipeline capable of initiating simultaneous execution on a variable number of instructions in a predetermined (i.e., known in advance) period of time. (See, Claim 1 lines 3-4.) Within this pre-established period of time, the logic sequencer can control the pipeline so that exactly a selected number of instructions enters the pipeline and begins execution (Claim 1 lines 11-13.) To re-state, there is some "predetermined" period of time that is known in advance, and which will be used to measure variable pipeline activities. A programmable count value can be selected so that instruction execution is initiated on exactly the number of instruction indicated by the programmable count value within that known period of time. To re-phrase yet another way, while the predetermined period of time is fixed in advance, the number of instructions executed within the time period is variable, and may be precisely controlled. This is clearly described by Applicants' Claim 1, which describes circuitry...

> "...to initiate concurrent execution on, the *predetermined number of the instructions in the predetermined period of time...*"

wherein that predetermined period of time is defined in Claim 1 as a predetermined time during which execution may be initiated on a variable number of the instructions within the pipeline, and the predetermined number of instructions is set by the count value. (Claim 1 lines 3-4, 7-8 and 11-12.) As discussed above in reference to Applicants' Figures 10-17, in Applicants' exemplary embodiment, the "predetermined period of time" is six clock cycles, although in other embodiments, other timeframes may be used.

The Examiner states that this capability is shown by setting a threshold to 1. The Schroter predetermined time period is then the amount of time required for an instruction to "progress". As best understood from this statement, the Examiner is stating that an exemplary period of time in Schroter is that time required for an instruction to begin execution within the pipeline when the threshold value is set to "one". Let's assume this is the case. That is, all threshold values are set to "one", then the time for an instruction to begin execution within the pipeline is measured. For Schroter to teach Applicants' invention, there must be some way in Schroter to re-adjust the threshold value(s) so that in this *same measured time period*, the pipeline can be controlled so a *different* programmable number of instructions begins execution in the pipeline. For example, by setting one or more of the threshold values to "two", two instructions now begin execution within the pipeline in the measured time period instead of the one instruction that began execution with these threshold values were set to "one".

Nothing in Schroter teaches or suggests the type of operation described above. As previously described, in Schroter, the execution units appear to be executing at full speed as often as possible. In fact, it is the very purpose of the Schroter system to maximize this full-speed operation. The threshold values do not affect this operation in any direct manner. If there is some indirect correlation between threshold values and the rate of execution of the execution units, the nature of correlation is not discussed in Schroter.

For at least the foregoing reasons, Schroter does not teach, or even suggest Applicants' Claim 1.

The remaining independent Claims 11 and 19 includes aspects of the invention similar to those discussed above in reference to Claim 1. For the reasons discussed above, these Claims are likewise not anticipated by Schroter, and this rejection should be withdrawn.

Finally, as best understood, the Examiner appears to state that based on prior arguments, Applicants agree that Schroter reads on Claim 1. (Final Action, page 4, lines 11-14.) While this argument is not understood, the conclusion is false. Applicants' Representative most certainly does not agree with that assertion for at least the reasons discussed above, and for those set forth in the previous response dated 1/15/2003.

2. Applicants' Representative appreciatively acknowledges the indication of allowable subject matter in Claims 2-10, 12-18, and 20-22. It is respectfully submitted that in view of the reasons set forth above, these Claims are allowable as presently presented.

CONCLUSION

Claims 1-22 were pending in the subject case. In the Final Action dated 2/25/03, Claims 1, 11, and 19 were rejected, and 2-10 and 12-18, and 20-22 were objected to. Claims 1-22 remain pending as previously presented. In view of the reasons set forth above, it is respectfully submitted the Claims are in condition for allowance. If the Examiner has questions or concerns regarding this response, a call to the undersigned is encouraged and welcomed.

Respectfully submitted,

Beth L McMahon 04/25/2003

Beth L. McMahon
Attorney for Applicants
Reg. No. 41,987
Tele No. (651) 635-7893

Unisys Corporation
M.S. 4773
P.O. Box 64942
St. Paul, MN 55164-0942

BLM

CERTIFICATE OF MAILING (37 CFR 1.8(a))

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail, in an envelope addressed to: Box NON-FEE AMENDMENT, Assistant Commissioner for Patents, Washington, D.C. 20231, on the date shown below:

Beth L McMahon 4/25/03 **April 25, 2003**

(Beth L. McMahon)) ***(Date)***